

IN THE CLAIMS

1. (Previously Presented) In an event processing server, a method for processing events comprising the steps of:
 - receiving an event message, the event message containing event registration information including product versions currently supported and a definition of a set of classes for said product, wherein said definition of a set of classes includes, for each class, a name, a unique identifier, a description of the class, and definitions of dynamic variables for each class, said dynamic variables including properties and alarm attributes;
 - identifying event information required to process event data based on the event message;
 - based on the event information, determining if existing event information is accessible to process the event data and if the existing event information is not accessible:
 - i) providing an event rejection indicating missing event information; and
 - ii) receiving the missing event information identified in the event rejection.
2. (Original) The method of claim 1 further comprising the steps of:
 - selecting the event information based on the event data received;
 - and
 - generating an event output from the selected event information.
3. (Canceled)
4. (Original) The method of claim 1 wherein the event message includes at least one unique identifier identifying the source of the event data.

5. (Original) The method of claim 4 wherein the step of identifying event information required to process event data identifies the event information required based on the source of the event data.
6. (Original) The method of claim 1 wherein the event message includes at least one unique identifier identifying event information required to process the event data.
7. (Original) The method of claim 1 wherein the steps of receiving comprise a step of accepting at least one of event registration information, event data and event information mark-up language documents.
8. (Original) The method of claim 1 wherein the event data includes network management data indicating a network management event associated with a source of the event data and wherein the step of receiving event data utilizes a hypertext transport protocol to receive the event data.
9. (Original) The method of claim 1 wherein in the step of determining, if the existing event information is accessible, the method further comprises the steps of:
 - i) providing an event data destination; and
 - ii) receiving the event data via the event data destination.
10. (Original) The method of claim 9 wherein the steps of receiving comprise the steps of:

reading first and second event data;

processing the first and second event data to produce event output data that reflects a hierarchical event relationship between the first and second event data.

11. (Original) The method of claim 1 further comprising the step of creating system component status records and wherein the step of receiving the event data further includes the step of:
 updating a status of the system component status record based on the event data received.
12. (Original) The method of claim 1 wherein the event message contains event data.
13. (Original) The method of claim 12 wherein the event message contains event registration information.
14. (Previously Presented) In an event generation client, a method for processing events comprising:
 sending event registration information including identifying event information required to process event data, said event registration information further including product versions currently supported and a definition of a set of classes for said product, wherein said definition of a set of classes includes, for each class, a name, a unique identifier, a description of the class, and definitions of dynamic variables for each class, said dynamic variables including properties and alarm attributes;
 detecting an event;
 in response to detecting an event, creating event data; and
 sending the event data to an event processing server.
15. (Original) The method of claim 14 wherein the step of creating event data includes formatting the event data in a mark-up language format capable of transmission via a hyper-text transport protocol.

16. (Previously Presented) The method of claim 14 wherein the step of sending event registration information further comprises the step of:
 - initiating a multiple of status checks of sources to produce status check information; and
 - forwarding status check information in the event data to the event processing server.
17. (Previously Presented) The method of claim 14 wherein the step of sending the event data further comprises the step of:
 - periodically sending event data to the event processing server as confirmation of an operating communications channel.
18. (Original) The method of claim 14, further including the steps of:
 - receiving an event rejection indicating missing event information from an event process server;
 - obtaining the missing information; and
 - sending the missing event information to the event processing server.
19. (Previously Presented) An event processing server for processing event messages comprising:
 - a memory;
 - a communications interface;
 - a processor; and
 - an interconnection mechanism coupling the memory, the processor and the communications interface;

wherein the processor is configured to:

 - receive an event message, the event message containing event registration information including product versions currently supported and a definition of a set of

classes for said product, wherein said definition of a set of classes includes, for each class, a name, a unique identifier, a description of the class, and definitions of dynamic variables for each class, said dynamic variables including properties and alarm attributes;

identify event information required to process event data based on the event message;

based on the event information, determine if existing event information is accessible to process the event data and if the existing event information is not accessible:

i) provide an event rejection indicating missing event information; and

ii) receive the missing event information identified in the event rejection.

20. (Original) The event processing server of claim 19 wherein the event processing server is further configured to:
select the event information based on the event data received; and
generate an event output from the selected event information.
21. (Canceled)
22. (Original) The event processing server of claim 19 wherein the event processor is further configured such that the event message includes at least one unique identifier identifying the source of the event data.
23. (Original) The event processing server of claim 22 wherein the event processor, in identifying event information, is further configured to identify event information required to process event data identifies the event information required based on the source of the event data.

24. (Original) The event processing server of claim 19 wherein the event processor is further configured such that the event message includes at least one unique identifier identifying event information required to process the event data.
25. (Original) The event processing server of claim 19 such that in the step of receiving the event processing if further configured to:
accept at least one of event registration information, event data and event information mark-up language documents.
26. (Original) The event processing server of claim 19 wherein the event processing server is further configured such that the event data includes network management data indicating a network management event associated with a source of the event data and wherein the step of receiving event data utilizes a hypertext transport protocol to receive the event data.
27. (Original) The event processing server of claim 19 wherein if the existing event information is accessible, the event processing server is further configured to:
 - i) provide an event data destination; and
 - ii) receive the event data via the event data destination.
28. (Original) The event processing server of claim 27 wherein, in the steps of receiving, the event processing server is further configured to:
read first and second event data;
process the first and second event data to produce event output data that reflects a hierarchical event relationship between the first and second event data.

29. (Original) The event processing server of claim 19 wherein in the steps of creating system component status records and the step of receiving the event data the event processing server is further configured to:
update a status of the system component status record based on the event data received.
30. (Original) The event processing server of claim 19 wherein the event processing server is configured such that the event message contains event data.
31. (Original) The event processing server of claim 30 wherein the event processing server is configured such that the event message contains event registration information.
32. (Previously Presented) In an event generation client, comprising:
 - a memory;
 - a communications interface;
 - a processor; and
 - an interconnection mechanism coupling the memory, the processor and the communications interface;wherein the processor is configured to:
send event registration information including identifying event information required to process event data, said event registration information further including product versions currently supported and a definition of a set of classes for said product, wherein said definition of a set of classes includes, for each class, a name, a unique identifier, a description of the class, and definitions of dynamic variables for each class, said dynamic variables including properties and alarm attributes;
detect an event;

in response to detecting an event, create event data; and send the event data to an event processing server.

33. (Previously Presented) The event generation client of claim 32 wherein, in the step of creating event data, the event generation client is further configured to include formatting the event data in a mark-up language format capable of transmission via a hyper-text transport protocol.
34. (Original) The event generation client of claim 32 wherein, in the step of sending, the event generation client is further configured to:
 - initiate a multiple of status checks of sources to produce status check information; and
 - forward status check information in the event data to the event processing server.
35. (Original) The event generation client of claim 32 wherein, in the step of sending, the event generation client is further configured to:
 - periodically send event data to the event processing server as confirmation of an operating communications channel.
36. (Original) The event generation client of claim 32 wherein the event generation client is further configured to:
 - receive an event rejection indicating missing event information from an event process server;
 - obtain the missing information; and
 - send the missing event information to the event processing server.
37. (Previously Presented) A computer program product that includes a computer readable medium having instructions stored thereon such that,

when the instructions are carried out by a communications device, the communications device is capable of performing the steps of:

receiving an event message, the event message containing event registration information including product versions currently supported and a definition of a set of classes for said product, wherein said definition of a set of classes includes, for each class, a name, a unique identifier, a description of the class, and definitions of dynamic variables for each class, said dynamic variables including properties and alarm attributes;

identifying event information required to process event data based on the event message;

based on the event information, determining if existing event information is accessible to process the event data and if the existing event information is not accessible:

- i) providing an event rejection indicating missing event information; and
- ii) receiving the missing event information identified in the event rejection.

38. (Previously Presented) An event processing server, for event processing, comprising:

- (i) a memory;
- (ii) a communications interface;
- (iii) a processor;
- (iv) an interconnection mechanism coupling the memory, the processor and the communications interface;
- (v) means, coupled to the communications interface, for receiving an event message, the event message containing event registration information including product versions currently supported and a definition of a set of classes for said product, wherein said definition of a set of classes

includes, for each class, a name, a unique identifier, a description of the class, and definitions of dynamic variables for each class, said dynamic variables including properties and alarm attributes;

- (vi) means, coupled to the communications interface, for identifying event information required to process event data based on the event message;
- (vii) means, coupled to the communications interface, based on the event information, for determining if existing event information is accessible to process the event data and if the existing event information is not accessible:
 - i) providing an event rejection indicating missing event information; and
 - ii) receiving the missing event information identified in the event rejection.

39. (Previously Presented) The method of claim 1 wherein said event registration

information includes information that identifies a source of forthcoming event data as well as event information that the event processing server will require in order to be able to correctly process the forthcoming event data.

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Previously Presented) The method of claim 1 further comprising maintaining a time history of a series of events related to an alarm attribute.
44. (Previously Presented) The event processing server of claim 19 wherein the server is further capable of maintaining a time history of a series of events related to an alarm attribute.
45. (Previously Presented) The method of claim 43 further comprising tracking how often the attribute is down and how often the attribute is up based on said time history of a series of events related to an alarm attribute.
46. (Previously Presented) The event processing server of claim 44 wherein the server is further capable of tracking how often the attribute is down and how often the attribute is up based on said time history of a series of events related to an alarm attribute.
47. (New) The method of claim 1 wherein the event information is separate from the event data, the event data defining how the event data is processed.
48. (New) The method of claim 47 wherein the event information includes XML templates.
49. (New) The event processing server of claim 19 wherein the event information is separate from the event data, the event data defining how the event data is processed.